CI102

LAB 7

Objectives

1. Identify the entities of a detailed design for your project.
2. Create detailed designs for a subset of the entities of your project.
3. Assess your team's technical capability compared to the technical needs of the project.

Designers need to specify the details of the entities that make up the system. These definitions should be sufficiently detailed that the design can be given to a developer and the developer can create the entity as envisioned by the designer.

Once your team starts to develop a design, you should also be developing a better understanding of the technologies and skill levels needed to build the product. As a separate task, this lab will also provide a chance for you to assess your team's capability to work on the project and identify learning or skill development you may need.

Procedure

Step 1 – Draft a list of entities for your project

You should consider the following types of entities:

- Screens (or Web pages)
- Database tables
- Files (e.g., data that is stored as part of the system but not stored in a database)
- Code (modules, objects, or functions)

Use Figure 7-1 to list all the system entities that you can identify. A good way to start is to pick one area and focus on that. For example, if your system has a significant user interface, start by trying to name all the screens that would comprise your interface. For each entity you list:

- Enter a type, e.g., "screen"
- Give it a meaningful name, e.g., "CustomerProfile"
- Provide any short notes or explanation needed to identify the screen, e.g., "This screen captures customer information and preferences."

Step 2 – Create detailed designs for at least 4 of your entities.

You will not be able to design all the entities of your system in this lab, but this step will get you started. Pick 4 entities that you think you understand the best at this point, and create a design for them. Every entity should have a name, type, and design details. Templates are provided to help you create detailed design for screens, database tables, and code functions.

Step 3 – Review your detailed designs.

After creating your designs, review them for completeness and clarity. Ask yourself this question: "If I was the developer and a designer handed me this design, would I know what to build without needing to ask a lot of questions?

If you have created the design entities as a team, set them aside for a few minutes before review each one. If you have worked in sub-groups within your team to create the designs, then exchange designs so the reviewer is a different person than the creator of a design.

Revise your designs based on the review.

Step 3 – Assess your team's capability to complete this project.

Once you have an architectural overview and the beginning of a design, you should be able to assess capability and identify things that someone on the team may need to learn. Use Figure 7-5 to summarize this information.

3.A – List the technologies you need for your project using the column on the left. Consider things such as programming languages, operating systems, specialized data sources, software libraries, support tools, and hardware.

3.B – List each team member at the top of a column, and then evaluate that person's knowledge of the technology in each row. For the column for each team member, use the following values:

1 – No knowledge or not much relative to the needs of this project
2 – Enough knowledge to accomplish part but not all of this project
3 – Knowledge probably sufficient for this project

3.C – Discuss within your team how you will start to gain capabilities that you are missing. You do not need to turn in results of this discussion in this lab, but will need to address this in the coming weeks.

What to Turn In

In order to obtain full credit for this lab, ***each team*** must turn in:

1. Figure 7-1 – Possible System Entities
2. Detailed designs for at least 4 entities in your system. Use the templates in Figures 7-2 through 7-4 to get started.
3. Figure 7-5 – Team Capability Assessment

Figure 7-1 – Possible System Entities

Product: Social Hour
Team: 85
Date: 2/24/17

Type	Name	Description or Notes
screen	Main Menu	Displays live feed
screen	Add/Edit Event	Allows user to add or edit events with field name and field values
screen	Settings	User can manage application and profile
screen	Calendar	Allows user to view calendar by the day, week, or month
screen	Friends	Allows user to view friend list and add friends
screen	Group	Allows user to create, view, and join groups
database	MongoDB	User data will be stored, MongoDB will be interfaced with RoboMongo
database	Google	User calendar would be stored using Google Calendar
files	Cache	
function	create_group	Function that creates an initially empty group with a specific set of attributes
function	establish_friendship	Function that establishes a friendship between two people objects
technoligies	Android Studio	used to program the application
technologies	GNU Image Manipulation Program (GIMP)	used to design the user interface and the logo of the application
technologies	Inkscape	used to design the icon and vector manipulation

CI102

ENTITY 1:
Type: Screen

Name: Main menu screen

Purpose: This screen serves as the home page for the application, meeting requirements 3, 4, and 5, which are the mobile interface, social networking, and stationary header for the application.

Description: Figure 1 shows the layout for this screen. This screen allows users to scroll down their feed of group and friend activity and access the other features of the application

The screen contains the following elements:
Calendar button: This element will allow the user to click the calendar button to access the calendar.
Menu bar: This element allows users to switch between the five main views : Friends, Groups, Main Page, Add/Edit Event, and Settings, in that order.

Layout:

Figure 1: Main Screen

ENTITY 2:
Name: establish_friendship

Type: Function

Purpose: This function is needed to meet requirement 4, which is the social networking portion of the application.

Parameters: The following parameters are used to call this function:

Name	Data Type	Notes
friend1	friend object	passed by reference
friend 2	friend object	passed by reference

Return Type: boolean - returns true if the friendship connection succeeded, returns false if failed

Processing:

```
pseudocode:
bool establish_friendship(friend &friend1, friend &friend2)
     if(!friend1.is_friends_with(friend2))
          friend1.friends_list.push_back(friend2)
          friend2.friends_list.push_back(friend1)
          return true
     else
          return false
```

ENTITY 3
Name: create_event

Type: Function

Purpose: This function is needed to meet requirement 4, which is the event creation and social media portion of the design.

Parameters: The following parameters are used to call this function:

Name	Data Type	Notes
event_name	string	
is_all_day	boolean	
event_time	integer	empty of is_all_day is true
group_affiliation	group	Potentially can be empty
event_date	string	
friends_invited	vector<&friends>	Potentially can be empty individual friends passed by reference if necessary
privacy	integer	0 if public, 1 if limited to friends, 2 if limited to best friends

Return Type: event object
Processing:

```
pseudocode:
bool create_event(string event_name, bool is_all_day, int event_time,
group
                  group_affiliation, string event_date, vector<&friends>
                  friends_invited, integer privacy)
     if invalidDate(event_name) throw exception //cancel event creation
     if invalidTime(event_time) throw exception; //cancel event creation
     return event(event_name, is_all_day, event_time, group_affiliation,
                  event_date, friends_invited, privacy) //return the event
                                                        //for use in java
                                                        //code
```

CI102

ENTITY 4:

Type: Screen

Name: Create Menu screen

Purpose: This screen is needed for requirement 3, the mobile interface of Social hour.

Description: Figure 4 1 shows the layout for this screen. This screen allows users to edit their preferences for the event before creating the event, partially using the create_event function.

The screen contains the following elements:

Calendar button: This element will allow the user to click the calendar button to access the calendar.
Menu bar: This element allows users to switch between the five main views : Friends, Groups, Main Page, Add/Edit Event, and Settings, in that order.
Event date: number box (textbox with restrictions to integers) that allows user to put event date in the application.
Is all day: checkbox for the user to indicate if the event takes place all day.
Event time: number box that allows users to input the time of the event. Input auto-formats (including placing the colon in the middle). Greys out if "Is all day" is checked.
Add Group: textbox that is actually a button, shows dropdown of groups the member is a part of. When added, the group is inserted into the textbox as a string.
Add Friends textbox that is actually a button, shows dropdown of the member's friends. When added, the friend is inserted into the textbox as a string.

Layout:

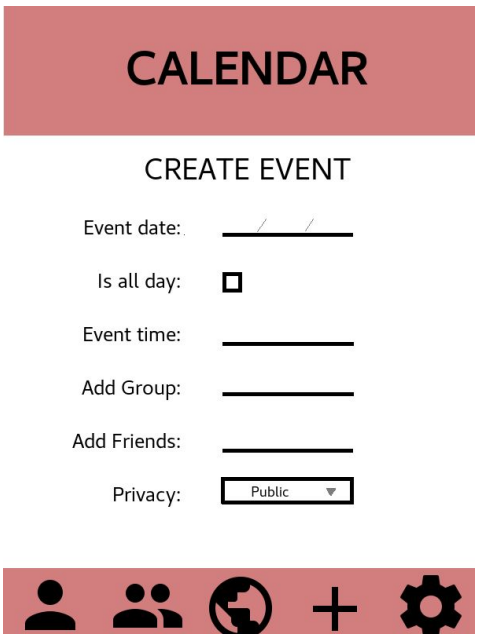

Figure 7-4: Create Event Screen

Figure 7-5 – Team Capability Assessment

Capabilities	Michael	Rocco	Dylan	Gavin
Ability to Access and Manage the Server	3	3	3	3
Ability to Program the Android Application	3	2	3	2
Ability to design the UX	1	2	2	3
Ability to perform data analytics on the database content	2	2	2	3
Ability to lead the team in a substantial direction	3	2	2	2

** The table values represent an assessment of team member capabilities. The values are:

1 – No knowledge or not much relative to the needs of this project
2 – Enough knowledge to accomplish part but not all of this project
3 – Knowledge probably sufficient for this project